

段取りの導出を行う調理支援システムの提案

高野 哲郎*¹ 上島 紳一*²

要 旨

料理を作る際、調理者はその段取りを考える必要がある。しかし、レシピに記述されている段取りは、調理者個々の環境に基づいたものではない。また、各手順においても調理場所や時間など、種々の経験的な了解などが要求されるため、初心者にとって必ずしも理解し易いとは言えない。

本稿では、調理者毎の異なる環境における段取りの導出法を提案する。まず初めに、インターネット上に存在するレシピデータを収集し、構造解析と情報の補完を行い、調理に必要な情報を明確にした調理シナリオの形に変換する。次に、調理シナリオから調理の段取りに必要な値を抽出し、調理における制約を考慮したスケジューリングを行って段取りを導出する。最後に、本稿で提案する調理支援システムのプロトタイプシステムを示す。

Proposal of a User-Support System that Provides Cooking Procedures

Tetsuro TAKANO Shinichi UESHIMA

Abstract

This paper discusses a cooking support system, "Cooking Studio", for cooking novices. Our system first collects recipes from the Web, analyzes them morphologically and semantically to generate, a Cooking Scenario. We derive information from Web recipes and add data which is necessary for scheduling. Next we describe a scheduling method to derive cooking procedures based on each user's cooking environment. We also show a prototype system of "Cooking Studio". Our system displays cooking procedures expressed in a time chart as the result of scheduling in the user's own kitchen; as well as instructions, figures, and movies. We believe "Cooking Studio" provides helpful GUI for cooking novices.

*¹株式会社 NTT データ関西

*²関西大学総合情報学部

1. はじめに

近年、Web上には様々な料理のレシピが存在している^{[1][2]}。レシピには料理に必要な材料と調理法が簡潔に記載されている。しかし、経験や知識が不足している料理初心者にとっては、調理工程を想像するのに十分な情報を与えているとは言えない。

また、レシピに記載されている調理手順は個々の調理者が調理を行う環境に基づいたものではない。そのため、レシピには調理場所や道具など制約がある場合や複数の料理を並列的に調理する場合における調理の段取りに関する情報が含まれていない。

料理初心者はこれらのレシピに含まれない情報を読み取ることが困難であるため調理の失敗に陥り易いと考えられる。そこで我々は、この問題の解消策として初心者が想起しづらい情報をシステムが提示することで調理を支援するアプローチを考えている。本稿では、初心者に対して「調理の手本」を示す調理支援システム Cooking Studio の提案を行う^[3]。ここでは主に本システムで用いる調理の段取りの導出法に述べ、本研究にて試作したプロトタイプシステムについて触れる。

本稿では、本システムの処理の流れに沿って説明を行う。まず初めに、調理工程を明示的に示す情報として調理シナリオを定義する。調理シナリオは、本システムが記述内容通りに動作することで調理をシミュレートすることを目標とした XML 文書である^[4]。

ここでは、本システムが対象データとして扱う Web 上のレシピデータ（以下、Web レシピと記す）を収集し、レシピの構造解析を行い、システムにとって調理の基本的な指示書となる調理シナリオの生成手順を述べる。また、ここではレシピに字面では記述されていない情報の補完を行う。このようにしてレシピの構造を明確化することで、スケジューリングや他にも調理シナリオの構造を用いた検索、情報の共有などが行いやすくなる。

次に、与えられた調理環境における調理の段取りの導出法について述べる。ここでは、前述の調理シナリオ上の単純工程を単位として調理スケジューリングを行う^[5]。本手法では、複数の調理シナリオから調理グラフを生成する。調理グラフは単純工程間の前後関係が表現可能な有向グラフとなっている。次に、調理特有の制約を定義し、調理における並列処理等を考慮した資源割付アルゴリズムを提案する。これにより、調理時間の短縮を図り、料理の特性に応じた調理の段取りの発見が可能となる。

最後に、本研究で作成した調理支援システム Cooking Studio のプロトタイプシステムについて述べる。我々は調理支援システムを「調理の手本」として利用することを想定している。まず、システム利用者は調理を行う環境を入力し、実際に調理を行う料理を選択する。次にシステムは入力された調理環境に基づいて調理の段取りを導出する。ここでは、得られた段取りを時間軸上にマッピングして表示する。

2. 研究動機

2.1 料理レシピの特徴

一般に、調理を行う際に参照するデータはレシピである。レシピは料理を作るために必要な材料とその調理法についての情報が簡潔に書かれている。調理法を簡潔に書くための方法として、レシピでは調理の動作を表すために料理特有の用語を用いられている。

このような簡潔に記述されている情報を理解するためには、背景知識が必要となってくる。つまり、レシピを読む調理者の経験や知識量によってはその内容を理解できない場合がある。よって、レシピの字面上の情報のみでは、調理を行うための全ての情報が表されているとは言いがたい。

また、レシピには調理の際に使用する調理器具や道具についても記述されている。しかし、これは調理者毎の異なる調理環境を考慮した内容ではない。つまり、レシピは個々の調理環境に応じて「いつ、どこで、どの料理についての調理を行えばよいか」という調理の段取りに関する情報を与えていない。そのため、調理者自身が個々の調理環境で実現可能な調理の段取りを決定しなければならない。

2.2 調理支援システム

2.2.1 要求仕様

調理経験や知識の少ない料理初心者がレシピの情報を解釈することや調理の段取りを決めることは困難である。この事は、初心者が料理の失敗に陥り易い原因にもなっている。そこで我々は、初心者がこれらの問題を克服するためのアプローチとして調理支援システムの利用を考えている。調理支援システムの要求仕様を以下に示す。

- 調理環境に応じた調理の段取りの導出

初心者が自力で調理の段取りを決めることが困難であるため、システムによって調理者が実際に調理を行う環境に基づいた段取りの導出を行う。

- 段取りを導出するためのレシピの詳細化

レシピは簡潔に書かれているため、調理法についての情報が不足している。また、レシピは個々の調理環境における調理スケジュールリングについて明示的に書かれていない。支援システムは上記の内容について不足している情報を追加する機能を持ち、レシピについて詳細な情報を把握する必要がある。

- Web 情報システムの実現

調理支援システムでは Web レシピを対象として扱う。その理由としては、利用者の利便性を考慮した点、ならびにデータの共有性が挙げられる。

また、Web 上のレシピは通常、視覚的に静的なデータとして存在している。調理支援システムでは、これを調理の様子を時間経過と共に動的に表現することで利用者にわかり

やすい情報の提示を行う必要がある。

2.2.2 支援方法

本稿では、上記の調理支援システムとしての要求仕様を満たすために下記の方法を用いてシステムを構築し、初心者への支援を行う。

(1) 調理シナリオの定義と Web レシピからの生成

本稿では、調理の各動作を明確に表すために XML を用いた構造化文書として調理シナリオを定義する。調理シナリオの生成には、まず、Web レシピの構造分析を行い、レシピの構造を理解することで不足している情報を特定し、マニュアル操作などで適宜補完を行う。調理シナリオを生成する利点としては、第一に後述する調理スケジューリングに利用できる点が挙げられる。また、それ以外にも、文書の共有や構造を利用した検索等の再利用が考えられる。

(2) 調理段取りの導出

本システムでは、実際に調理者が調理を行う環境に基づいた調理手順を導出する。本手法では、まず調理工程の情報をラベル付エッジとして持つ有向グラフを用いて表現する。そして、調理における基本的な制約を定義し、グラフのパスに着目した4.4で述べる調理に必要な資源割付アルゴリズムにより調理の段取りを導出する。ここでは調理時間の短縮を目指し、複数の動作を並列処理することで調理を段取り良く行える調理スケジューリングを行い、その結果をタイムチャートの形で示す。

(3) 調理支援システムの構築

本システムでは、調理シナリオより導出した調理の段取りを時間順に表示する。ここでは、マルチメディアコンテンツを利用して調理のシミュレーションを行い調理の具体的な様子を表現する。シミュレーションでは、タイムチャートに示すそれぞれの調理工程の具体的な様子を表す画像や映像、また説明のために文章を画面に表示する。このように、本システムでは調理の段取りと動作の様子を組み合わせる画面に表示し、利用者にとって直感的に理解し易いインタフェースを与える。

前述の要求仕様と支援方法の対応関係を図1に示す。

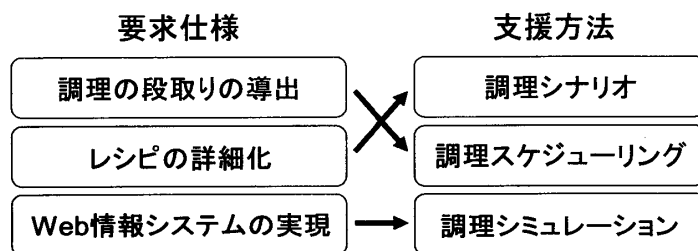


図1 要求仕様と支援方法

2.3 先行研究

本システムの先行研究として以下のものが挙げられる。濱田ら^[6]は料理教材をもとにした調理手順の構造解析手法を提案している。ここでは、料理の調理法について辞書を用いて解析し、結果をグラフで構造化表現している。また、これらは料理番組に付随した教材テキストを対象としており、料理映像とレシピの対応付けることでマルチメディア料理教材として統合することを目的としている^[7]。

これに対して、本システムでは、Web上のレシピデータを対象としており、調理工程についてのスケジューリングを行った結果を用いて段取りと調理動作の様子を表現することを目的としたシステムとなっている。上記の研究とは調理工程を調理シナリオというXML文書により明確に示す点、調理手順の導出を目標としている点が異なる。

3. 調理シナリオ

本節では、本システムにとって調理の基本的な指示書となる調理シナリオの定義とその生成法について述べる。調理シナリオでは、レシピを料理についての情報、材料についての情報、調理法についての情報と3つの部分に大別して、それぞれについて詳細な記述を行える文書構造を設ける。また、調理シナリオは後述する調理スケジューリングで利用するために、段取りの導出を行うために必要な情報を記述できる形式にしている。

調理シナリオは、図2に示すようにWebレシピを元データとして段階的に生成を行う。まず初めに、Webレシピによって書かれている調理動作の詳細を表現する方法として、XMLを用いた構造化文書である調理シナリオを定義する。この調理シナリオは調理動作を中心に据えた文書スキーマに即して記述を行う。ここでは、Webレシピに対して形態素解析を行い、文書スキーマを基に各要素情報に対して意味付けを行うことで、Webレシピから調理シナリオへの変換を図る。

次に、調理シナリオに対して情報の補完を行う。ここでは、文書スキーマでは要素として定義されているがWebレシピの字面上の情報として含まれていない情報についてマニュアルによって値を追加する。

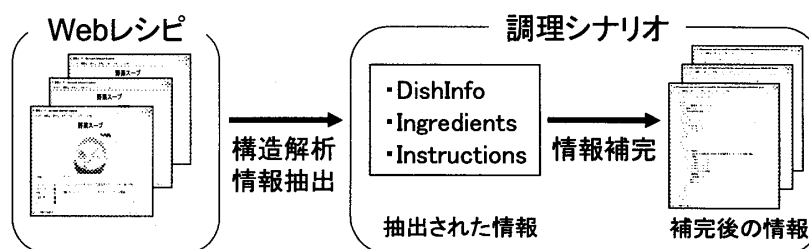


図2 調理シナリオの生成手順

3.1 調理シナリオの定義

調理シナリオは、Web レシピが持つ情報を明確に記述するために用いる文書形式を指す。一般に、レシピで書かれている調理法は調理動作を中心とした記述されていることが多い。例として図3（出展^[8]）を示す。また、下線語は調理動作を示す語であることを表す。

本手法ではこの性質を活かして、調理動作を中心に据えた調理シナリオの文書スキーマを定義する。この文書スキーマは各調理動作において扱われる材料や、調理の制約に関係のある調理場所や器具等の記述が行えるような構造を持つ。次に、レシピから文書スキーマの各要素に該当する情報を抽出し、それらを値としてもつ文書を調理シナリオとする。

料理レシピの記述内容から調理シナリオの文書スキーマを図4のように定義する。このスキーマでは、まず最上位要素として調理シナリオ〈Cooking Scenario〉を持つ。料理一品の調理法について書かれている調理シナリオには、この要素が1つだけ存在する。

里芋の煮っころがし（調理法）

1. 里芋は水で洗い皮をむく。
2. 鍋にだし汁、酒、みりん、しょうゆと1の里芋を入れて煮る。

Taro Tumbles (Instruction)

1. Wash the taros in cold water, and peel.
2. Place the stock, sake, mirin, sugar, soy sauce and taros in a pot, and simmer.

図3 レシピと調理法の記述形式

```

<CookingScenario>::=<DishInfo>,<Ingredients>,<Instructions>
  <DishInfo>::=<DishName>,<Type>,<Image>
  <Ingredients>::=<Number>,<Ingredient>,<Ingredient> | <Ingredients>,<Ingredient>
    <Ingredient>::=<Name>,<Amount>,<MaterialUnit>
  <Instructions>::=<Instruction> | <Instructions>,<Instruction>
    <Instruction>::=<Unit> | <Instruction>,<Unit>

<Unit>::=<UnitID>,<Text>,<Action>,<Material>,<Description>,<Place>,<Utensil>,
  <Tool>,<Time>
  
```

図4 調理シナリオの文書スキーマ

文書スキーマ中の調理シナリオ要素では次の3種類の情報を子要素として持ち、それぞれの要素は下記のような構造となっている。

- 料理に関する情報 〈DishInfo〉

料理に関する情報は、料理名 〈DishName〉、料理の種類 〈Type〉 と料理の完成イメージ 〈Image〉 に関する情報を指す。

- 料理を作るのに必要な材料 〈Ingredients〉

材料には何人分の材料であるか 〈Number〉 とどの材料 〈Ingredient〉 を扱うか、そして、各々の材料は材料名 〈Name〉 と数量 〈Amount〉 と単位 〈MaterialUnit〉 が書かれている。

- 調理法 〈Instructions〉

調理シナリオでは Web レシピに書かれている調理法の文章を調理動作を示す動詞毎に区切り、その単位ごとに動作の内容を記述する。ここでは、その単位を単純工程と呼び、調理シナリオでは単純工程を調理法における最小単位としている。

調理法はレシピでは段落に区切って書かれていることが多いので、この段落を調理工程 〈Instruction〉 として扱う。さらに、調理工程には複数の動作が含まれているため、そこから単一の動作を粒度とした工程、つまり単純工程 〈Unit〉 に細分化する。各単純工程は、“どの材料を使い 〈Ingredients〉、どの容器に入れて 〈Utensil〉、場所はどこで 〈Place〉、どの道具を使い 〈Tool〉、どれだけの時間・どのタイミングで 〈Time〉、どのような 〈Description〉 調理をする 〈Action〉” という情報が含まれている。そして、“単純工程自身を表す ID 〈UnitID〉 を持ち、どの単純工程の後すべきか 〈Premise〉” という情報と、“レシピ中でのその該当する文章 〈Text〉” もあわせて持つ。

3.2 情報補完

ここでは、Web レシピに不足している情報の補完について述べる。Web レシピには文書スキーマで定義された全ての要素についての情報が含まれていない。この不足している情報は、料理に関する知識を用いることで導くことができるため、レシピ文中では省略されている。

調理シナリオでは後述する調理スケジューリングで扱うため、全ての要素について値を持つ必要がある。暗黙的なデータを明示的に示すには、Web レシピの字面を解析しても情報としては不十分である。文書スキーマで定義されている要素のうち不足している情報を補完する必要がある。

3.1から、調理法の最小単位は単純工程であり、この単純工程において不可欠な情報は動作であると言える。また、材料以外の情報は調理動作に基づいて判断できるため、レシピ中では省略されている場合が多い。本稿では、この動作以外の情報を材料とそれ以外の情報（場所、道具、時間等）に区別し、それぞれの情報を補完する方法について以下に述べる。

3.2.1 中間生成物の表現による材料の補完

材料は調理工程を経ることにより、調理開始前に用意された状態から徐々に変化していく。調理シナリオでは、変化する材料の様子を把握する仕組みを持つ。

まず、各単純工程において材料を調理した結果、つまり調理における中間生成物をその単純工程の ID を用いて表す。また、調理シナリオではこの中間生成物を別の単純工程で扱う材料として記述することを許す。これにより、単純工程間の前後関係を導くことができ、ついでには調理全体の流れを把握することが可能となる。

【例】「じゃがいもは皮をむいて、2 cm 角に切ります」という調理工程を例に説明する (図 5)。この工程は、「皮をむく」という単純工程 “Unit 1” と「2 cm 角に切る」という単純工程 “Unit 2” という 2 つの単純工程から構成されている。まず、“Unit 1” で扱う材料は「じゃがいも」であるので、その調理の結果として「皮をむいたじゃがいも」が生成される。そこで、この “Unit 1” の調理結果を [Unit 1] と記述する。

次に、“Unit 2” で扱う材料は、工程文中では明記されていないが、文脈より「皮をむいたじゃがいも [Unit 1]」であると判断できる。よって、“Unit 2” の材料は [Unit 1] とする。“Unit 2” によって「2 cm 角に切った [Unit 1]」が生成される。これは「皮をむいた後に 2 cm 角に切ったじゃがいも」であることを表す。

また、“Unit 2” の材料に [Unit 1] を扱うことから、“Unit 2” を行う前に “Unit 1” を行う必要があることが分かる。このように、中間生成物を表記する仕組みによって単純工程間の関係が明らかとなる。

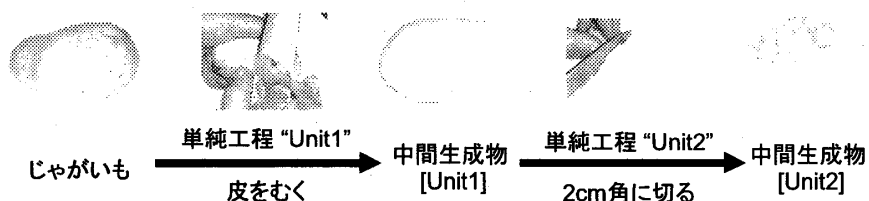


図 5 中間生成物の表現

3.2.2 動作と材料以外の情報の補完

単純工程において調理器具や道具、場所や時間等といった (動作と材料以外の) 情報は、その調理動作から判断できる。調理シナリオでは上記の情報が記述されていない場合は、動作から導き出される値を補完すべき情報とする。

3.3 調理シナリオの生成

ここでは、Web データであるレシピから調理シナリオを生成する方法について述べる。

3.3.1 レシピからの要素の抽出

まず、レシピから調理スキーマの要素である〈DishInfo〉、〈Ingredients〉、〈Instructions〉を抽出する。要素の抽出には、レシピが書かれている HTML 文書中の出現位置を利用する。図 6 に野菜スープのレシピから要素を抽出する様子を示す。

3.3.2 料理情報の解析

3.3.1で抽出した要素のうち、料理自身についての情報を分析する。調理スキーマより〈DishName〉、〈Type〉、〈Image〉を抽出する。ここで、〈Type〉は料理名や調理法などから判断し、「煮物」「炒め物」「パスタ料理」等の値を入れる。図 6 の場合では、〈DishName〉は「野菜スープ」、〈Type〉は「煮物」、そして〈Image〉は「画像ファイルの URL」を抽出する。

3.3.3 原材料 ID の割当

3.3.1で抽出した要素のうち、材料について分析を行う。まず、各材料に対して原材料 ID を与える。次に、調理スキーマに従って、〈Name〉、〈Amount〉、〈MaterialUnit〉を抽出する。図 6 の Ingredients より要素を抽出した結果を表 1 に示す。

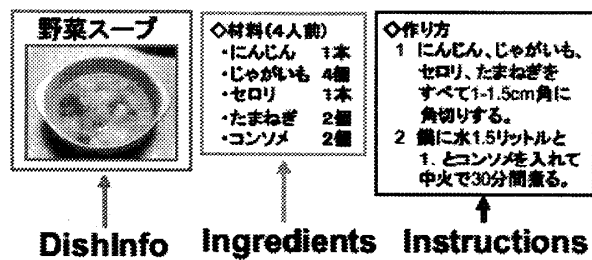


図 6 レシピからの要素の抽出 (3.3.1)

表 1 原材料 ID の割当 (3.3.3)

原材料ID	Name	Amount	MaterialUnit
m1	にんじん	1	本
m2	じゃがいも	4	個
m3	セロリ	1	本
m4	たまねぎ	2	個
m5	コンソメ	2	個
m6	水	1.5	リットル

注：m6 は 3.3.4 に割当られる

原材料 ID は調理により加工されていく材料の状態を把握するため、「その材料は調理開始前に準備された状態の物である」ことを示す。また、前述のように中間生成物は単純工程の ID を用いて表す。調理シナリオでは材料をこの 2 種類の記法（すなわち、原材料 ID と中間

生成物)を用いて表現し、調理法の最小単位である単純工程間の前後関係を導く。原材料 ID は m_i (i は自然数の添え字) と記述する。

3.3.4 単純工程の導出法

次に、調理法から単純工程を導出する。調理法の最小単位である単純工程毎に、調理法に関する情報を抽出する。また、単純工程において不可欠な情報は調理動作であることから、調理動作を基準として単純工程を導出する。

● 単純工程における調理動作の抽出

調理スキーマより、図6にある〈Instructions〉の各段落〈Instruction〉に対して形態素解析を行い、調理の動作を表す「動詞」の基本形を単純工程〈Unit〉の〈Action〉とする。

調理スキーマの要素である〈Action〉は動詞、〈Place〉、〈Utensil〉、〈Tool〉、〈Time〉は名詞である。よって、これらの要素を導くために〈Instruction〉を入力文として形態素解析を行う^[9]。

● Cooking Table の生成

Cooking Table は表2で示すような各単純工程に含まれる情報のリストである。調理シナリオの文書スキーマに記述された単純工程の要素となる情報を抽出し、それをCooking Tableの値とする。なお、このCooking Tableは後述する補完を行い、全ての項目に対して値が格納された後で、調理シナリオ生成の際に利用する。

Cooking Tableの値にはレシピに記載されている情報のみを利用する。これは、単純工程における調理動作の抽出する際に行った形態素解析の結果から表の該当部に言葉を埋める。Cooking Tableのそれぞれの項目の値に関しては以下のように抽出する。

表2 Cooking Table (3.3.4-3.3.5)

UnitID	Action	Ingredients	Description	Place	Utensil	Tool	Time
Unit1	切る	m1,m2,m3,m4	1-1.5cm角に	(調理台)	(まな板)	(包丁)	(*3)
Unit2	入れる	Unit1,m5,m6	(*1)	(調理台)	(鍋)	(*2)	(*3)
Unit3	煮る	(Unit2)	(*1)	(コンロ)	(鍋)	(*2)	30分

(): 補完によって得られた情報 *1: 修飾語が空値

*2: 道具がいない場合は空値 *3: 動作に相応な時間を表す

「材料」〈Ingredients〉

単純工程で扱われた材料を原材料 ID または中間生成物を示す単純工程の ID を用いて表す。ここで、調理法の文中には材料で書かれていなかった食材が登場する場合がある。その食材についても3.3.3と同様に原材料 ID を割り当てる。

図6の〈Instructions〉中の2つめの〈Instruction〉を例に挙げる。文中の「1.」は前

の〈Instruction〉で加工した材料全体を表している。よって、この単純工程の扱う材料には中間生成物 [Unit 1] を値として与える。

単純工程の動作の内容によっては材料が2つ以上必要な場合もある。例えば、ある単純工程で「混ぜる」という動作がある場合、そこで扱う材料は2つ以上あると考えられる。そのため、「m1を混ぜ」と混ぜる材料が1つしか書かれていない場合には、原則としてその直前の単純工程による調理結果、つまり中間生成物の値を加えることとする。

「修飾語」〈Description〉

「動作」に関する修飾語をこの部分に記入する。

「場所」〈Place〉

調理場の中で動作を行う場所、もしくは機械に関する名詞を記入する。

「器具」〈Utensil〉

加工する材料を入れたり置いたり包んだりする物を表す名詞をこの要素として抽出する。

「道具」〈Tool〉

調理するために使う道具についての名詞を抜き出す（手の場合もある）。

「時間」〈Time〉

単純工程中の時間（例：20秒）、もしくはその動作を行う時間的条件（例：きつね色になるまで）を書く。「まで」を含む文節、助動詞「たら」の入った文節などはこの項目に記入する。

最後に、残った単純工程句中の不必要な言葉（読点・句点や助詞・助動詞等）を削除する。上記の要素に該当する言葉は数百数千語も存在しないと予想される。そのため、事前に辞書を用意しておけば自動判別もある程度可能であると考えられる。

3.3.5 調理法に関する情報の補完

Cooking Table の値は調理工程に関する最小限の情報と定義している。そのため、調理を行うにはこの表の全項目に関する情報を完全に把握する必要がある。

ここで補完すべき情報とは Cooking Table 中で空値となっている項目である。ここでは、3.2の手法に基づきそれぞれの項目が空値の場合に情報を補完する流れについて述べる。

「材料」の補完

何らかの調理を行うということは必ず1つ以上の材料を扱うので、どの単純工程においても「材料」は空値であってはならない。しかし、単純工程で扱う材料はその文中に明示的に書かれていない場合が多くある。この場合、3.2.1より中間生成物を単純工程で扱う材料とする。

材料として扱う中間生成物は以下のように特定をする。例えば、材料について記述がない単純工程“UnitN”の動作を表す動詞が連用形であれば、直前の単純工程“Unit (N-1)”が何らかの関係を持つ、つまり連続して調理が行われていると判断する。よって、“UnitN”では中間生成物 [Unit (N-1)] を扱う材料とする。

「修飾語」の場合

修飾語は空値でも構わない項目である。

「場所」「器具」「道具」の補完

それぞれの値は「動作」と「材料」によって決定する。場所に関しては、主として焼いたり煮たりする場所は「コンロ」、食材を水で洗う場所は「流し」となる。また、器具や道具は「切る」という動作の場合、器具には「まな板」を、切る道具には「包丁」を用いる。

「時間」の補完

動作に相応な時間がかかる、という意味の値を補う。

情報補完の例として、表2ではCooking Tableに情報補完により与えられた値を()を用いて表す。

3.3.6 調理シナリオの生成

3.3.1から3.3.4で得られた値や3.3.5で補完を行ったCooking Tableの値を基に調理シナリオを生成する。調理シナリオは図4に示す文書スキーマに従い、XMLを用いて記述を行う。

3.4 まとめ

本節では、Webレシピに明示的に含まれていない調理に関する情報を追記できる文章である調理シナリオの定義とその生成法について述べた。調理シナリオの特徴としては以下の2点が挙げられる。

● 情報を明確に記述する

Webレシピを単純工程ごとに詳細を記述することで、各単純工程の情報を明確にした。また、材料の中間生成物を表現することにより、単純工程間の前後関係を導くことが可能となった。

● 構造化文書の再利用性

暗黙的な情報を明示的に示している調理シナリオは、例えば、次節で述べる調理スケジュールリングでの利用や、構造情報を利用した検索が可能であるため、生成元となったWebレシピよりもデータとして利用範囲が広がっている。

4. 調理の段取りの導出

本節では、実際に調理者が調理を行う環境に基づいた調理の段取りの導出法について述べる。ここでは調理における基本的な制約を定義し、この制約を充たすスケジュールリング手法を提案する。

4.1 調理の制約

調理者は並列的に調理を行うことで効率化を図る場合や、複数の料理を調理する場合、様々な制約を考慮に入れながら段取りを決定し実際に調理を行う。本研究では、調理に関する制約

から以下に示す基本的なもののみを考慮に入れて段取りの導出を行った。

- 調理場所の制約

調理人数、コンロやオーブンの数が該当する。また、コンロやオーブンは並列処理が可能な場合がある。

- 時間の制約

各料理では調理をする工程の順番は決められている。また、料理によっては前後する工程を連続的に行う場合や、複数の工程について終了する時間の同期をとる場合も存在する。

- 料理自身の持つ制約

料理の種類によって、「煮物は最初に」、「パスタ料理は最後に」等の調理する順番を決める必要がある。

4.2 調理スケジューリング

ここでは、4.1で述べた調理の制約を充たす調理スケジューリング手法の提案を行う。本手法では、調理シナリオを用いて調理手順の決定に必要な情報を抽出する。調理シナリオには下記に示す特徴を持つ。

- 調理法を単純工程という単位を用いている
- 各単純工程の情報を詳細な記述が行える
- 単純工程間の前後関係を明示的に表現している

これらは調理スケジューリングを行うにあたって必要な情報となっている。以下、調理スケジューリングに必要な調理シナリオが実際に複数与えられているものと仮定する。

本手法では以下に示す2つの原則に従いスケジューリングを行う。1つは、調理時間の短縮を図るため、できるだけ短い時間で全ての料理の調理を行う。もう1つは、全ての工程を直列的に調理せず可能な限り並列的に調理を行う。

これらのポリシーのもとでスケジューリングを行い調理の効率化を図る。ここでいう調理の効率化とは、調理時間の短縮や調理者の労力の削減等を意味する。

次に、上記のポリシーに基づいて調理のスケジューリングを行うアルゴリズムについて述べる。このアルゴリズムでは、前節で得られた単純工程をスケジューリングにおける単位として扱う。これにより、与えられた調理環境における調理スケジューリングを導く。

段取りの導出は以下の手順で行う。まず、(1)調理シナリオより調理グラフを生成し、(2)グラフにおいてパスの分割を行う。次に、(3)パスに対して順序を割付け、(4)エッジに対して資源割付けを行う。最後に(5)調理の段取りを示すタイムチャートを生成する。

4.2.1 調理グラフの生成

調理グラフは単純工程についての情報を持つ調理シナリオから料理の各単純工程の前後関係を把握するために生成する。調理グラフの定義は以下の通りである。

定義 調理グラフはノードとエッジからなる非巡回的な有向グラフである。ノードは変化する

る材料の状態を表す。また、複数の料理を調理する様子をグラフで表現するため、Start ノードと Goal ノードを用いる。

調理グラフ全体から見て、先端である Start ノードは調理全体の開始時点、末端の Goal ノードは全ての料理が完成した状態を表す。Start ノードからは各料理の調理開始ノードへエッジを引き、各料理の完成状態を表すノードから Goal ノードへエッジを引く。

エッジは単純工程を表し、エッジのラベルには単純工程の内容を示す。有向エッジは2つのノードの前後関係を表し、方向は前方に位置づけられたノード（材料）から、調理を行い後方のノード（加工された食材）へ向かうものとする。

各エッジにはそのエッジが表す単純工程についての情報をラベルとして持つ。ラベルには(1)その単純工程は並列処理が可能であるか、あるいは直列処理を行うのか、(2)調理を行う場所、(3)調理時間に関する情報を記述する。

調理グラフの例を図7に示す。これは3つの料理についての調理シナリオを元に作成した調理グラフである。AからJのエッジによって単純工程を示し、各エッジのラベルにはその単純工程についての情報を記述する。

ここで、調理シナリオより調理グラフを生成する前に、調理グラフ上で表現する調理シナリオで記述されている単純工程間の前後関係や各単純工程の情報を整理しておく必要がある。これらの情報は、調理グラフ上でエッジが持つラベルの情報やエッジ同士の接続関係を示すため調理グラフの生成には欠かせない情報である。

そこで、調理シナリオから調理グラフを生成する前準備について述べる。ここでは、調理シナリオから調理グラフのエッジのラベルに関する情報をまとめ、表を作成する。調理グラフの生成には、エッジ間の接続関係を示す情報とエッジのラベルで表す情報が必要となる。これらの情報は全て調理シナリオから抽出する。

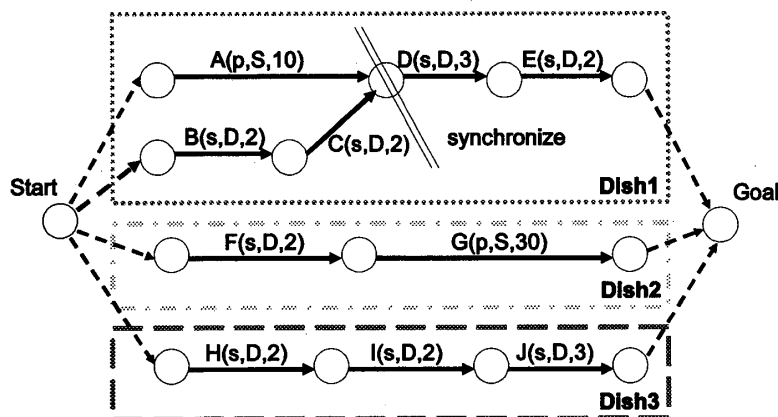


図7 調理グラフ (4.2.1)

まず、エッジ自身を示す情報を導き出す。一本一本のエッジは、調理シナリオにおいては単純工程を表す。また、調理グラフにおいて各エッジを識別するために ID を割り振る必要がある。そのため、一度に調理を行う全ての料理に対して、単純工程を一意に表す EdgeID を割り振る。

次に、各料理においてエッジの前後関係を把握する。この前後関係の導出には、調理シナリオの Ingredients を利用する。単純工程 “UnitN” の Ingredients に材料の中間生成物を表す UnitID（ここでは [UnitM] とする）が含まれているとする。この時、単純工程 UnitN は単純工程 UnitM によって生成されるものを材料として扱うため、UnitN は必ず UnitM の後に調理を行わなければならない。このルールを用いて、それぞれの単純工程を調理する前に行うべき単純工程の EdgeID を Premise の値とする。

そして、エッジのラベルで表す情報を取り出す。初めに、単純工程における調理動作の特徴 Feature を抽出する。調理動作が「煮る」「茹でる」等の並列的な調理が可能な場合、並列処理を表す “parallel” の値を代入する。また、Premise の値として複数の EdgeID を持つ時、それぞれの単純工程の終了時間の同期をとる必要がある。このことを値として “synchronize” を代入する。上記の種類には該当しない単純工程は直列的に調理を行うので、値として “series” を代入する。次に、調理を行う場所 Place と時間 Time の値をそれぞれ取り出す。

調理グラフの生成例として、表 3 に示す 3 品の調理法を示す調理シナリオから調理グラフを生成する場面を考える。まず調理シナリオからエッジのラベルに関する情報をまとめた表を生成する。表 4 は調理グラフの生成に必要なエッジのラベルに関する情報 (EdgeID, Premise, Feature, Place, Time) を調理シナリオから抽出しまとめた表である。

エッジのラベルに関する情報では以下の項目について値を持つ。

[DishNum]

調理する料理の ID を表す。例えば、表 3 で書かれている料理の順に Dish 1, Dish 2 … と ID を付与する。

[EdgeID]

全ての料理において単純工程を一意に示す ID を与える。Dish 1 の最初の単純工程から順に A, B … と ID を付与する。

[Premise]

その工程を調理する直前に行うべき工程を表す。調理シナリオで材料として扱う中間生成物から以下のように導く。

【例】表 3 の料理名 “スパゲティー”，工程 ID “3” の単純工程を例に挙げて説明する。この単純工程で扱う材料は “Unit 2, m 4” と書かれている。これは、Unit 2 で調理を行った材料を Unit 3 で扱うことを意味する。つまり、Unit 3 で調理を行うためにはその直前に Unit 2 を行う必要がある。この関係より、表 [表 3 \(tbl:schedule2\)](#) の工程 ID “C” では、先行工程 “B” を値として持つ。

表3 調理シナリオの例 (4.2.1)

DishName : スパゲティー

UhitID	Ingredients	Action	Place	Time
Unit1	m1	茹でる	コンロ	10
Unit2	m2,m3	切る	調理台	2
Unit3	Unit2,m4	混ぜる	調理台	2
Unit4	Unit1,Unit3	混ぜる	調理台	3
Unit5	Unit4,m5	和える	調理台	2

DishName : スープ

UhitID	Ingredients	Action	Place	Time
Unit1	m1	切る	調理台	2
Unit2	Unit1	煮る	コンロ	30

DishName : サラダ

UhitID	Ingredients	Action	Place	Time
Unit1	m1,m2	切る	調理台	2
Unit2	Unit1	混ぜる	調理台	3
Unit3	Unit2,m3	和える	調理台	3

表4 エッジのラベルに関する情報 (4.2.1)

DishNum : Dish1

EdgeID	Premise	Feature	Place	Time
A	-	Parallel	Stove	10
B	-	series	Dresser	2
C	B	series	Dresser	2
D	A,C	synchronize	Dresser	3
E	D	series	Dresser	2

DishNum : Dish2

EdgeID	Premise	Feature	Place	Time
F	-	series	Dresser	2
G	F	Parallel	Stove	30

DishNum : Dish3

EdgeID	Premise	Feature	Place	Time
H	-	series	Dresser	2
I	H	series	Dresser	2
J	I	series	Dresser	3

[Feature]

調理を行う場所が「コンロ」「オープン」、動作が「煮る」「茹でる」等の並列的な調理が可能な場合、この項目に“parallel”の値を代入する。また、複数の先行工程がある場合は、それぞれの工程の終了時間の同期をとることを表す“synchronize”を値として代入する。上記に該当しない調理工程（表4中の値は“-”）では、直列的に調理を行う。

[Place]・[Time]

単純工程を行う場所や時間を表す。ここでは、調理シナリオに書かれている値を代入する。

最後に、エッジについての情報から調理グラフを生成する。工程表の先行工程よりノードとエッジの接続関係を導く。調理グラフの各エッジのラベルには“工程 ID (時間, 場所, 調理の動作)”を記述する。また、動作の特徴である“synchronize”を表すために、そのエッジの始点ノードに二重線を引く。表 4 より生成された調理グラフは図 7 のようになる。

4.2.2 パスの分割

ここでは、4.2.1で作成した調理グラフを対象として、同じ料理において分岐がなくなるまでパスの分割を繰り返す。パスとは、調理グラフの連続するエッジにおいて最大の調理時間を含む分岐のない部分グラフを指す。パスの分割を行う際には、調理グラフから該当するエッジを除去する。そして、得られたパスはリスト化しておく。

パスの分割は、4.2.3に示す料理の種類によって調理を行う順序を割り付けるための前準備として行う。この順序を割り付ける単位としてパスを用いる。また、ここで部分グラフ化を行う理由は、同一料理において並列化して調理を行う場合にどのパスから調理を行えば良いかを計算するためである。図 7 内の Dish 1 においてパスの分割の結果を図 8 に示す。

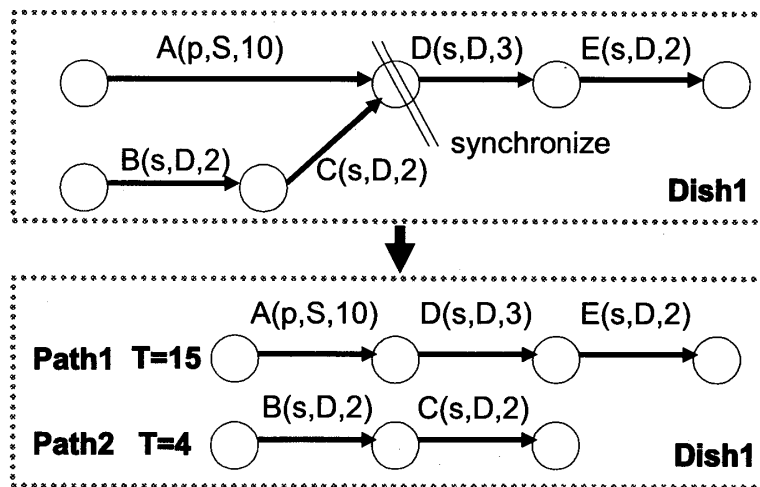


図 8 調理グラフのパス分割 (4.2.2)

4.2.3 順序割付

4.2.2で得られたパスに対して料理の種類に応じた調理を行う順序を割り付ける。ここでは、調理の段取りを決定する際に資源を割り付ける優先度を料理単位で付与する。順序割付のルールを以下に示す。

[順序割付ルール]

- 料理の種類により調理の順番を決める

それぞれの料理を一番おいしい状態で食べることができるよう、料理にいくつかの種類

を分けて、種類によって調理の順番を設定した。

● 調理時間が長い料理から先に料理を行う

同じ種類の料理が複数ある場合は、調理時間が長い料理（あるいはパス）から先に料理を行う。これは、先に完成する料理がもう一方が完成するまでに待つ時間を短くするためにこの処理を行う。

【例】4.2.2で得られたパスに対して順序割付を行う。ここでは、料理の種類を「煮物・冷し物」「anytime」「焼き物・炒め物」「麺類（パスタ）」の4種類とし、調理の順番をこの通りとする。これは、それぞれの料理を一番おいしい状態で食べられるようにそれぞれの順番を設定した。「煮物・冷し物」は、高・低温状態で長時間調理を続けるために一番初めに調理を行う。一方、麺がのびてしまうため「麺類（パスタ）を扱う料理」は全調理の最後に調理する。また、「焼き物」「炒め物」は料理が温かいままで食べるために「麺類」の前の順番とした。最後に、これら以外の種類の料理は調理時間を選ばないため、その種類を「anytime」とした。

図7にある Dish 1・Dish 2・Dish 3 は、表3よりそれぞれスパゲティー・スープ・サラダの調理グラフを表しており、それらの調理の種類は麺類・煮物・anytimeとなる。これに順序割付ルールを適応すると、パスの優先度は Dish 2・Dish 3・Dish 1となる。順序割付を行ったパスのリストを図9に示す。

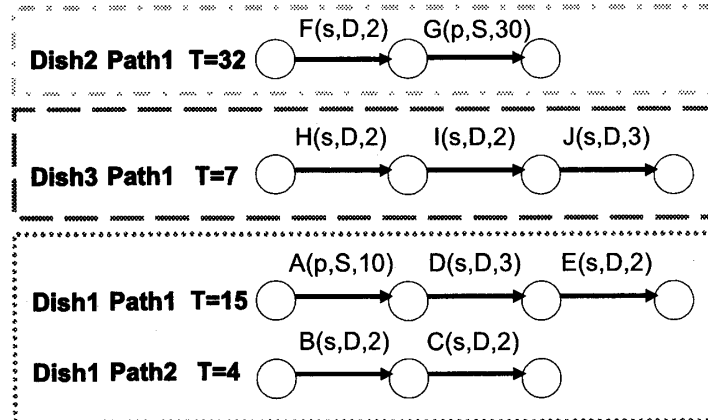


図9 順序割付後のパスリスト (4.2.3)

表5 エッジに資源を割り付ける優先度 (4.2.4)

DishNum	PathNum	FdgeID		
Dish2	Path1	F	G	
Dish3	Path1	H	I	J
Dish1	Path1	A	D	E
Dish1	Path2	B	C	

4.2.4 資源割付

4.2.3で割り付けたパスの順序に基づき各エッジに対して資源を割り付ける。資源とは、調理における資源として「調理開始時間」「調理終了時間」と「調理場所」を指す。これにより各単純工程は「調理開始時間」「調理終了時間」「調理場所」という調理手順に関する情報が取得できる。本手法で用いる資源割付アルゴリズムを以下に示す。

[資源割付アルゴリズム]

A 1. エッジの優先度を示す表を作成する

エッジに資源を割り付ける優先度についてまとめた表を作成する。4.2.3で求めたパスの優先度(表5)を基に、縦方向にパスの優先度順、横方向はエッジの前後関係について値をまとめた表を表5のように作成する。

A 2. 調理環境における調理場所を特定する

個々の調理環境において存在する調理場所(「調理台」「コンロ」「オープン」)を特定する。それぞれの調理場所は「現在使用中」か「使用可能」のどちらの状態であることを示す値を持つ。

A 3. 時間Tにおいて調理可能なエッジを探す

時間T(調理開始時点から経過した時間、初期値=0)において、(1)先行工程がない、もしくは全て終了している、(2)既に調理開始時間が与えられていない、(3)パスの優先度が一番高い、これら3つの条件を満たすエッジがあれば、A 4.へ進む。全てのエッジがこのルールに該当しない場合、全てのエッジに調理開始時間が与えられていれば終了する。他に調理開始時間が与えられていないエッジがあれば、一番早く未使用となる調理場所が使用可能となる時間T'まで時間Tを進める。

A 4. 調理場所の使用状況を確認する

A 3.で求めたエッジのラベルに書かれている調理場所が時間Tにおいて使用可能であれば

A 5.へ進む。調理場所が使用中であればA 3.に戻り、他の調理可能なエッジを探す。

A 5. 並列処理が可能であるかを判定する

A 4.で求めたエッジに調理開始時間Tを、調理終了時間として時間Tにエッジに書かれた調理時間t分を加えた時間 $T+t$ を与える。また、このエッジが並列的に調理を行えるならば、調理場所を使用中として時間Tを1分進めてA 3.へ戻る。直列的に行う調理工程であれば、時間Tをt分進めてA 3.に戻る。

【例】コンロの数が2台という調理環境を想定し、図9で得られたパスリストに対して上記の資源割付アルゴリズムを用いた。その結果として得られた単純工程毎の調理開始・終了時間をまとめた表を調理スケジュールとして表6に示す。

4.2.5 調理時間の修正

ここでは、4.2.4でエッジに与えられた「調理開始時間」ならびに「調理終了時間」について修正を行う。

4.2.4では、「全ての単純工程は資源に空きがあればすぐに調理を行う」というスケジューリ

ングポリシーのもとで調理手順を決定した。しかし、上記の方法では同期を取るべき単純工程において、その対象となる単純工程の終了時間が揃わないため対応できない。

また、「麺類（パスタ）」の料理を調理する際にはその料理が完成する時点で全部の調理が終了することが望ましいが、前述の調理手順の決定方法ではこの条件を満たすことができない。

これらの問題点を解消するために、全料理の終了時間から開始時間へと遡って上記の条件を満たすように修正を行う。

【例】例として表6の調理スケジュールにて該当する単純工程について調理時間の修正を行う。ここでは、同期を取るべきエッジは調理終了時間が遅い方に揃える。また、「麺類」の料理のエッジについては、可能な限り調理終了時間に料理が完成するように遅い時間へずらす。調理時間の修正した結果を、表6では矢印を用いて示す。

4.2.6 調理タイムチャートを生成

4.2.5でエッジに与えられた「調理開始時間」「調理終了時間」「調理場所」を視覚的に表すために調理タイムチャートを生成する。調理タイムチャートは、調理場所と調理時間をそれぞれ縦軸、横軸にとり、各エッジを該当する位置に配置する。同期を取るべき単純工程は、縦線を引くことでそれを表す。表6より生成した調理タイムチャートを図10に示す。

表6 調理スケジュール (4.2.4-4.2.5)

EdgeID	調理開始時間	調理終了時間	調理場所
F	0	2	Dresser
G	2	32	Stove1
H	3	5	Dresser
I	5	7	Dresser
J	7	10	Dresser
A	10 → 17	20 → 27	Stove2
B	11 → 23	13 → 25	Dresser
C	13 → 25	15 → 27	Dresser
D	20 → 27	23 → 30	Dresser
E	23 → 30	25 → 32	Dresser

注：矢印左が修正前、右が修正後の調理時間を表す (4.2.5)

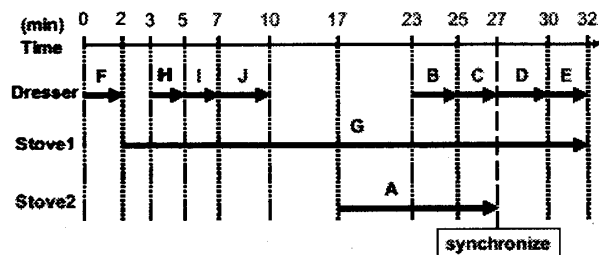


図10 タイムチャート (4.2.6)

4.3 検討

前述の調理スケジューリング法では、調理動作の特徴として並列と同期という2つの性質に着目した。これらについての有効性を示すために比較検討を行う。

まず初めに、並列的な調理を行わない場合を考えてみる。表3にある3品の料理の調理シナリオを用いて全ての調理動作を直列的に行った場合、その調理時間は58分となる。これに対して、本手法で得られる段取り(図10)通りに調理を行えば32分で全ての料理を完成させることができる。これにより、本手法では調理時間を短縮し効率的な段取りを導いていることがわかる。

次に、同期を考慮せずに調理時間の補正を行わない場合について考察する。図11は前述の4.2.5に示す調理時間の補正を行わずにタイムチャートを生成したものである。この場合、全体の調理時間は本手法で得られる段取りと変わらないが、出来上がる料理に差が生じると考えられる。補正を行わない場合、例えば麺類の調理を早く行ってしまいうために出来上がった状態では麺が伸びきってしまう、ということが考えられる。またこれは、4.1に示す3番目の制約を充たしていない。本手法では、料理の出来上がりも考慮に入れた段取りを導出している。

最後に、表3に示す3品の料理をコンロの数が1台という調理環境で調理する場合の段取りを本手法において導出した結果を図12に示す。本手法では、調理環境を調理台とコンロに限定して考え、コンロの数によって異なる段取りを導出することができる。

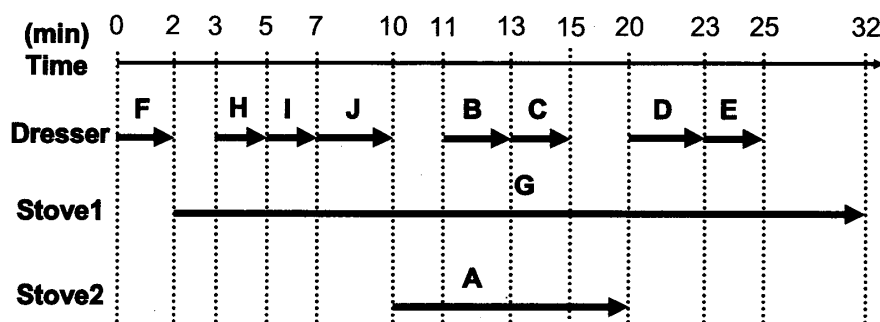


図11 同期を考慮しない場合

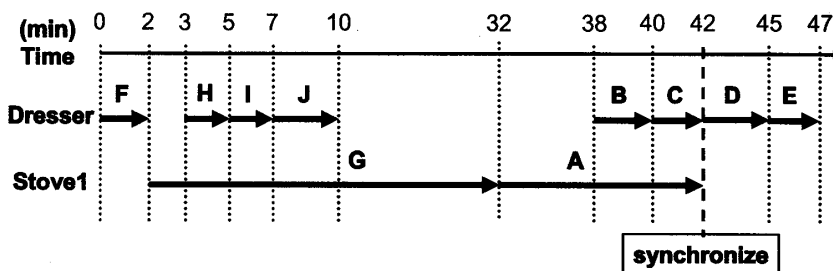


図12 調理環境が異なる場合

4.4 まとめ

本節では、与えられた調理環境における調理の段取りの導出法について述べた。本手法は基本的な調理の制約を充たすスケジューリングを行う。また、本手法には4.3で述べたような有効性があると考えられる。

5. 調理支援システム Cooking Studio

5.1 システム概要

ここでは、料理初心者が調理の際に直面する問題の解消を目的とした調理支援システム Cooking Studio を提案する。またここでは、Cooking Studio のプロトタイプシステムについて述べる。本研究では、初心者に対して「調理のお手本」を示し、初心者が調理をする際に利用できるシステムの実現を目指す。

本システムでは、調理シナリオより導出した調理の段取りを時間順に表示する。ここでは、調理シミュレーションと名づけて、タイムチャートに示すそれぞれの調理工程の具体的な様子を表す画像や映像、また説明のために文章を画面に表示する。このように、本システムでは調理の段取りと動作の様子を組み合わせる画面に表示し、利用者にとって直感的に理解し易いインタフェースを与える。

試作したプロトタイプシステムの構成を図13に示す。本システムの流れは以下の2通りである。

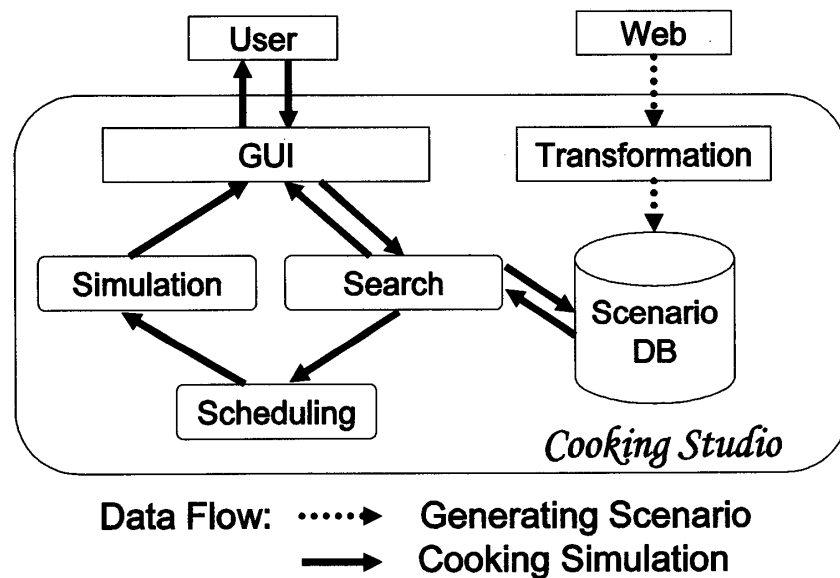


図13 システム構成図

- Generating Scenario

Web レシピを対象データとして扱い、調理シナリオを生成する。この際、3章で示したようにレシピの解析・構造化・情報補完を行い、調理シナリオを生成する。生成した調理シナリオは Scenario DB に格納する。

- Cooking Simulation

初めに、システム利用者が調理する料理を選び、その料理を調理する様子をシミュレートする。まず、Scenario DB から調理する料理に該当する調理シナリオを取得する。次に、調理シナリオを用いて調理を行う環境に応じた調理スケジューリングを行う。そして、本システムは調理スケジューリングにより得られた手順通りに調理シミュレーションを行い GUI に表示する。

5.2 プロトタイプシステム

前述した調理シミュレーションを行う調理支援システムのプロトタイプシステムの試作を行った。調理シミュレーションとは、システム上で利用者が選択した料理を作る際の段取りと単純工程ごとの調理の様子を表示することを意味する。

調理シミュレーションを行うに当たり、利用者は自分が調理を行う環境（本システムではコンロの数）についての情報を入力する。そして、調理をしたい料理をシステムの画面上で選択する。システムは利用者によって入力された料理に該当する調理シナリオを取得し、それを基に段取りの導出する。導いた段取りをもとに調理シミュレーションを GUI に表示し実行する。

調理シミュレーションの際に表示される情報について詳しく述べる。本システムの調理シミュレーション画面の例を図14に示す。図14の調理シミュレーション画面では、画面右側に調理の段取りを表示している。これは4章の調理スケジューリングによって得られるタイムチャート（図10）を基にしている。縦軸方向は時間を示し、上から下に向かって時間が進むことを表す。横軸方向は調理環境を表している。調理台やコンロといった調理場所毎に行うべき単純工程を矩形で示し、この矩形の中には各単純工程の調理動作を示す動詞を記述している。また、矩形の色によりどの料理に関する単純工程であるかを表している。

次に、画面左側は現在行うべき単純工程の様子を図や文字を用いて表示している。図14では、画面右側の段取りを示すところに「現時点」と書かれている単純工程についての情報を表示している。

今回はこのような形で調理シミュレーションを行うプロトタイプシステムの試作を行った。プロトタイプシステムの構築には macromedia Flash MX^[10] を用いて構築を行った。

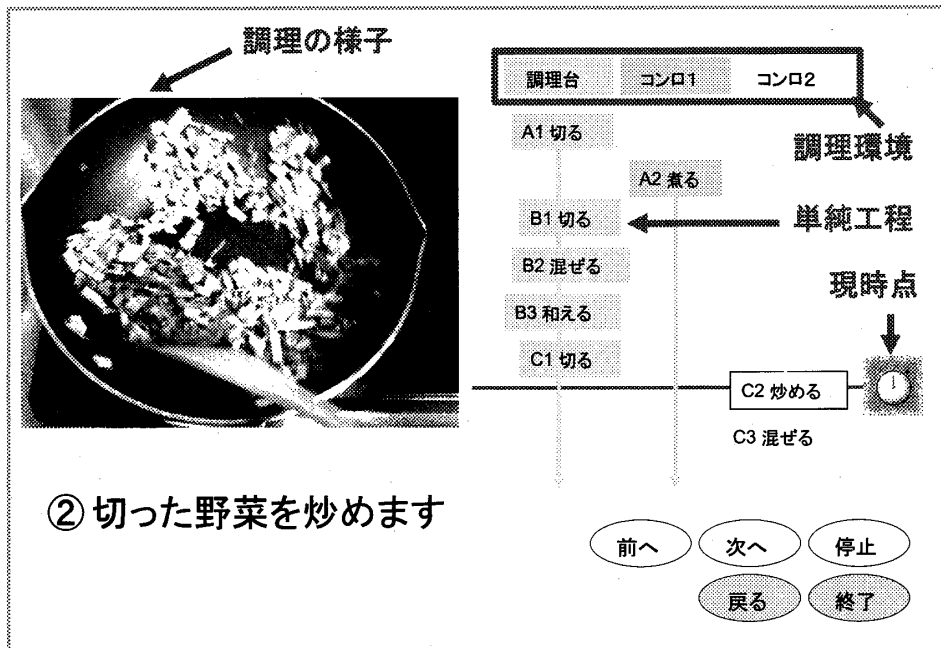


図14 システム画面 調理シミュレーション画面

6. 関連研究

関連研究としては、以下の研究が挙げられる。まず、濱田ら^{[6][7]}と本研究との比較を表7に示す。濱田らは、マルチメディア統合の一例として料理に関する辞書を作成しレシピのテキストデータの解析を行い、料理番組の映像と対応付けを行っている。本研究ではデータの再利用性を考慮に入れて文書スキーマを定義し、明示的に表現されていない情報を補完し調理シナリオの生成した。また、調理シナリオでは扱う材料の記法を定めることで調理の段取りの導出を可能とした。

梅原ら^[11]は内容や文書構造に類似性が見られるHTML文書群を、その性質を利用しXML文書への変換を行っている。ここでは、元の情報に対してXMLによって意味情報を付与している。その変換元のWebレシピ(HTML文書)に対して、意味情報を付与すると共に、Webレシピには含まれていない情報を補完して調理シナリオ(XML文書)を生成している。

表7 関連研究との比較

研究者	濱田ら ⁶⁾⁷⁾	本研究
調理過程の表現	グラフ表示	XML
目的	マルチメディア統合	スケジューリング
アプローチ	辞書の利用	材料の記法の導入
	動画処理	情報補完

一方、Web データの活用法として、灘本ら^[12]は Web 情報の受動的な視聴を目的とした Web 情報の番組化を行っている。ここでは、静的な Web 情報の文書構造を利用して動的な番組という形式に変換するアプローチをとっている。本研究では、Web データの文書構造を利用するだけでなく、文書が表す意味を解釈した上で必要な情報の可視化を行っている。

また、エージェントと利用者のインタラクション設計言語として石田ら^[13]が Q の提案を行っている。ここでは、設計言語 Q を 3D 仮想空間でのシミュレーションに応用している。本研究においても料理世界の言語として調理シナリオを定義し、Web 上にあるレシピデータをもとに調理シナリオを生成した。また、調理シナリオの解釈実行系として調理支援システム Cooking Studio の実現を目指している。

7. おわりに

本稿では、Web レシピを調理シナリオへ変換し、与えられた調理環境における段取りの導出を行う調理支援システムの提案を行った。料理初心者が調理の際に直面する問題に対し、本システムの利用によって情報の面での解消を目指している。本研究における今後の課題として、(1)HTML で記述されたレシピデータから調理シナリオの効率的な生成法、(2)より複雑な制約にも対応できる調理スケジューリングの改良、(3)調理支援システムの実用化などが挙げられる。

情報資源は将来において共有や再利用が可能な形で生成していく必要がある。今後、Web 適応技術の発展に伴って人と機械による協調作業を行う機会が増えると考えられる。

参考文献

- [1] 3分クッキング, <http://www.ntv.co.jp/3min/>.
- [2] おかずのクッキング, <http://www.tv-asahi.co.jp/okazu/>.
- [3] 高野哲郎, 上島紳一, “調理支援システム Cooking Studio の提案”, The Second International Conference on Creating, Connecting and Collaborating through Computing, 7B-3, 2004.
- [4] 高野哲郎, 上島紳一, “Cooking Scenario-レシピの Scenario 化とその応用-”, DBWS2003, 1A-4, 2003.
- [5] 高野哲郎, 上島紳一, “調理支援システム Cooking Studio: レシピのシナリオ化と調理スケジューリング手法”, DEWS2004, I-8-5, 2004.
- [6] 浜田玲子, 井手一郎, 坂井修一, 田中英彦, “料理テキスト教材における調理手順の構造化”, 電子情報通信学会論文誌, Vol.J85-D-II No.1, pp79-89, 2002年1月.
- [7] 三浦宏一, 高野求, 浜田玲子, 井手一郎, 坂井修一, 田中英彦, “料理映像の構造解析による調理手順との対応付け”, 電子情報通信学会論文誌, Vol.J86-D-II No.11, pp1647-1656, 2003年11月.
- [8] 藤田裕子, ナヴィ インターナショナル, “英語で作る和食”, ナツメ社, 2003.
- [9] 形態素解析システム 茶筌, <http://chasen.aist-nara.ac.jp/>.
- [10] macromedia FLASH MX, <http://www.macromedia.com/jp/>.
- [11] 梅原雅之, 岩沼宏治, 永井宏和, “事例にも基づく HTML 文書から XML 文書への半自動変換-シリーズ型 HTML 文書における類似性の利用-”, 人工知能学会論文誌, 16巻 5号 B, pp.408-416, 2001年.
- [12] 灘本明代, 服部多栄子, 近藤宏行, 沢中郁夫, 田中克己, “Web コンテンツの受動的視聴のための

自動変換とスクリプト作成マークアップ言語 “情報処理学会論文誌：データベース, vol.42, No.SIG1 (TOD8), pp.103-116, 2001.

- [13] 石田亨, 福本理人, “インタラクション設計言語Qの提案” 人工知能学会論文誌, vol.17 No.2, pp. 166-169, 2002.